

能動視覚エージェント群による 複数対象の協調的追跡

長尾 卓 浮田 宗伯 松山 隆司

京都大学大学院情報学研究科 知能情報学専攻

近年のコンピュータの発達に伴い、実世界を対象とするシステムの必要性が増してきている。動的に変化する実世界を把握する上で、人物や車両といった対象の運動を視覚的に捕え、追跡することは一つの有効な手段である。そのためには多様な状況に適應できる柔軟性が必要となるが、従来多用されてきた集中型のシステムでは、設計段階で想定されている状況以外の状況には対応しきれないことが多い。本論文では多様な状況変化に対して破綻しない対象追跡を実現するために、能動カメラを備えた計算機(能動視覚エージェント)を環境内に分散配置した分散協調型のシステムを提案する。本研究では、複数の能動視覚エージェントを環境内に配置し、それらの協調動作によって動的に変化する実世界を実時間で把握するシステムの設計を目的とする。特に、視覚的情報をもとにした複数対象の追跡のための協調方式を設計することに主眼をおく。

Multi-target Tracking by Communicating Active Vision Agents

T. Nagao N. Ukita T. Matsuyama

Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University

Detecting and tracking objects visually is one of the most effective ways to understand dynamic situations of real world environments. While computer systems working in the real world have to be flexible enough to adapt themselves to various situations, ordinary centralized systems can adapt only to predefined limited situations. In this paper, we propose a cooperative distributed system for tracking multiple moving objects. The system consists of a group of network-connected computers with active cameras (Active Vision Agent:AVA). The purpose of this paper is to design a dynamic interaction scheme among AVAs for tracking multiple moving objects.

1 はじめに

近年のコンピュータシステムの発達にともない、コンピュータシステムにより実世界環境を把握する必要性が増してきている。コンピュータビジョンを用いて実世界の状況を把握することは、遠隔講義システムなどの基盤技術であり、これからの発展が期待されている。実世界を把握するシステムの中でも、動きまわる対象を追跡することは環境の動的状況を把握するという意味で重要である。我々は、能動カメラを計算機に接続した能動視覚エージェント(Active Vision Agent, 以下 AVA)を複数台環境内に分散配置し、それらの中で協調することにより実世界の動的状況を把握する分散協調視覚システムに関して研

究を行っている。これまでに、AVAによる単一対象の協調追跡システムが提案されている。そのシステムは、協調により対象の3次元位置が得られるので対象が障害物により隠蔽された場合にも安定に追跡を継続できる。

対象が複数存在する場合には、単一対象の場合と異なり、対象の動きの多様性だけではなく、対象の数の変化の多様性にも対応しなければならぬ。具体的には、複数の対象が存在する場合にも対象の同定を安定に行わなければならない。対象の同定法として、高度な画像処理技術を用いて行う方法も考えられるが、本稿では、高度な画像処理技術に頼るのではなくAVA間の協調により対象の同定を安定に行い、複数対象の協調追跡を行うシステムを設計する。

2 システムの構成

2.1 能動視覚エージェント (AVA) による対象の追跡

能動視覚エージェント (Active Vision Agent, 以下 AVA) とは、視線方向を変えることのできる小型カメラを計算機に接続したものであり、計算機からの命令により、視線方向を変えたり、現在の視線方向を数値データとして得ることが可能である。また、背景差分アルゴリズムを用いることにより対象の検出を行うことができる。さらに、各計算機間はコンピュータネットワークにより相互接続されており、自分の検出した対象情報を他の AVA に伝えることができる。これらの機能を用いることにより、単独で、また、協調して対象を追跡することができる。これらの機能は、「知覚」、「行動」、「通信」という分類で整理できる。

AVA は自律性を持つものとしてモデル化されているため、単独の AVA でも対象の検出、追跡を行うことができる。AVA を用いた対象追跡の基本アルゴリズムは以下のとおりである (図 1)。

1. 視線方向を変化させながらあらかじめ対象世界を撮影した画像群から超広角パノラマ画像を合成する。
2. 現在のカメラの視線・ズーム倍率と一致したパン・チルト・ズームパラメータを持つ背景画像を超広角パノラマ画像から生成する。
3. 生成された画像と観測画像との差分をとり、変化のあった領域を移動対象として検出する。
4. 検出された対象が画像の中央で、適当な大きさに写るように、カメラの視線を制御する。

以上の処理を反復することにより、移動対象を自動的に検出、追跡するシステムを実現することができる。

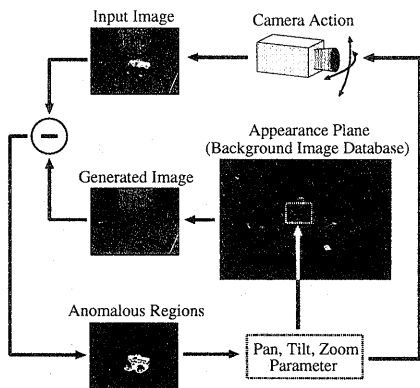


図 1: 対象追跡のアルゴリズム (文献 [1] より)

AVA の通信機能は、コンピュータネットワークを用いることにより、分散配置されているエージェント間で情報を交換し、対象の詳細な情報を得たり、他の

AVA とのメッセージを介した交渉を行ったりする。

AVA の通信機能に関しては、AVA の自律性を妨げないように非同期通信を基本とする。送信機能としては、一対一、ブロードキャスト通信を行うことができ、受信機能としては、メッセージバッファを設けることにより、自 AVA の動作状況にしたがってメッセージを到着順に調べることができる。メッセージを調べた時の AVA の動作状態によって、メッセージを処理する、メッセージを無視してバッファから取り除くというを行う。

AVA 間で送受信されるメッセージとしては、対象情報を基本としたメッセージと、協調のためのメッセージが存在する。また、メッセージの送受信は非同期が基本であるが、協調を行う上での意思交換を行わなければならない場合には、相手の AVA とメッセージを介した同期を行う必要がある。

ここで、本システムで用いられるメッセージをあげておく (表 1)。

MES.OBJECT.2D	カメラの撮影中心から検出対象の 重心に向かう視線ベクトル情報
MES.OBJECT.SDP	検出対象の 3 次元位置情報
MES.ID.FAIL	対象特定失敗
MES.AGENCY.IN	エージェントに参加する
MES.BYE	エージェントから退避する
MES.MASTER.NOMINATE	ノミネート要求
MES.NOMINATE.ACK	ノミネートする
MES.NOMINATE.NOT	ノミネートしない
MES.MASTER.CHANGE	マスタを変更する
MES.AVA.REQUEST	AVA 移動要求
MES.AVA.CANDIDATE	AVA 候補候補
MES.CANDIDATE.ACK	候補を受け入れる
MES.CANDIDATE.NOT	移動要求拒否、移動候補拒否
MES.AGENCY.UNIFY	Agency Unification を行い、 エージェントを統合する
MES.AGENCY.CHANGE	エージェント移動命令
MES.AGENCY.SPAWN	Agency Spawning を行い 新たなエージェントをつくる
MES.AGENCY.AMALGAM	Agency Amalgam を行い、 新たなエージェントに参加する
MES.AGENCY.SEGREGATE	Agency Segregation を行い、 新たなエージェントとして分離する

表 1: システムで用いられるメッセージ

2.2 エージェントによる対象の表現

複数の AVA を分散配置し、それらの間で協調処理を行わせる分散協調システムは、単独での追跡に比べて以下のような点で優れている。

- 対象の 3 次元位置を得ることができる。
- 他の AVA から送られてきた移動対象に関する情報を元に対象世界を探索し、すばやく対象を検出することができる。
- 障害物などにより対象が遮蔽された場合も他の AVA の情報を元に対象の出現を待ったり、別の対象を探索したりと、様々な動作パターンをシステムに追加することができる。

複数の AVA によって実世界内の移動対象を協調追跡する場合に、対象情報を AVA 間でどのように表現、共有するかという問題がある。本研究で設計するシステムでは、対象のシステム内表現としてエージェントを用いる。複数の AVA を用いて対象を追跡する場合には、同一の対象を検出、追跡している AVA のグループが形成される。そのグループをエージェントと呼ぶ。エージェントに属するすべてのメ

ンバは同一の対象を検出、追跡しているの、AVA間で対象を同一の情報として扱うことができる。また、エージェンシーを対象の表現として用いることによって、対象を検出しているAVAが一つかない場合でも表現が可能である。エージェンシーには、エージェンシー情報の管理、対象位置の算出および他のAVAに対象の3次元位置を通知するマスタAVAとその情報に従って対象を追跡するワーカAVAの2種類のAVAが存在する。また、エージェンシーに所属していないAVAはフリーランサーと呼ばれる。

システム内での対象の表現をエージェンシーとすることで、対象が実世界に現れ、動き回るということを、システム内でエージェンシーが動的に形成、継続するものとして扱うことができる。

2.3 システムの3つの階層

本稿で設計するシステムは、複数の自律的な処理主体が動的にインタラクションを行うように設計されている。AVA内では、知覚、行動、通信各モジュールが動的にインタラクションをしており、エージェンシー内では、マスタAVAとワーカAVAが動的にインタラクションしている。さらに、複数の対象が存在する状況では、エージェンシーも複数存在するのでエージェンシー同士の動的なインタラクションも考える必要がある。

このように、単一のAVA内のモジュール間のインタラクションと、エージェンシー内、エージェンシー間のインタラクションの3つのレベルでの動的なインタラクションを考えることができる(図2)。ここでAVA内のモジュール間インタラクションをIntra-AVA Layer、エージェンシー内のインタラクションをIntra-Agency Layer、そしてエージェンシー間のインタラクションをInter-Agency Layerとする。エージェンシー内、エージェンシー間の動的インタラクションは通信モジュールが中心となって行われる。

Intra-AVA Layerでは、知覚モジュールの知覚結果、行動モジュールから得られるカメラの視線方向、通信モジュールからの他のAVAが持つ対象情報などをどのようにして統合するのかということが問題となる。本研究では並行に動作する処理主体間での円滑な情報交換機構としてダイナミックメモリ [1][2]を用いる。ダイナミックメモリとは、並行動作する処理主体間に存在する時間拡張された共有メモリであり、並行動作する複数の処理主体から非同期に参照、記録することができる。ダイナミックメモリを用いることにより、あるモジュールは他のモジュールの動作、状態によらず必要な時刻のデータをいつでも読み書きすることができる。ダイナミックメモリとダイナミックメモリによるモジュールの動的統合については3.2で説明する。

Intra-Agency Layerではマスタとワーカの主従関係が主であり、やり取りされるデータはそのエージェンシーが追跡している対象に関する情報が基本とな

る。それに対し、Inter-Agency Layerではマスタとマスタの協調関係であり、やり取りされる情報としては、エージェンシーやそのメンバに関する情報が基本となる。エージェンシー間のコミュニケーションでは、マスタとマスタが通信する必要があるが、エージェンシー内でマスタが動的に入れ替わることを考えると、一対一の通信で常に2つのマスタ間をつなぐのは困難である。そのため、エージェンシー間のコミュニケーションに関しては、ブロードキャストを基本とする通信を用いる。ブロードキャストを基本として用いることにより、エージェンシー間のコミュニケーションの最中にマスタが入れ替わるということが起っても継続してコミュニケーションを行うことができる。

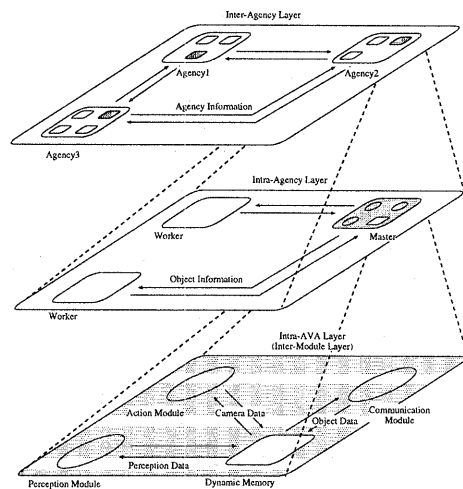


図 2: システムの3つの階層

協調の方法としては、以下のような協調プロトコルという形で設計する。

Intra-Agency Layerでのインタラクションは、次の3つのプロトコルによって実現される。

- Agency Formation** エージェンシーの生成消滅とエージェンシーへの参加脱退に関するプロトコル
- Role Assignment** エージェンシーを継続させるためのプロトコル
- Agency Spawning** 対象が二つに分離した場合にエージェンシーを分裂するプロトコル。

Inter-Agency Layerでのインタラクションは、次の4つのプロトコルによって実現される。

- Agency Unification** 対象同定の失敗などによって、一つの対象に対しエージェンシーが複数できてしまう状況に対処するプロトコル。
- Agency Amalgamation** 複数の対象が近接して分離検出できない場合に、エージェンシーを融合して対処するプロトコル。

Agency Segregation Agency Amalgamation 後に対象が分離検出できるようになった場合に、エージェントシーを分離させるプロトコル。

Agency Restructuring エージェント間のメンバー数の格差を小さくするためのプロトコル。

また、メッセージ以外でシステムの動作の流れを変更するものとしてイベントというものを考える(表2)。イベントは、メッセージの受信やAVAの動作結果などによって生成される。イベントもメッセージと同様にイベントキューに記録され、順次処理される。

Detect Object Event	対象を検出した
Lost Object Event	対象を見失った
Become Master Event	マスタになる
Become Worker Event	ワーカーになる
Become Free lancer Event	フリーランサーになる
Calc 3Dp Average Event	3次元位置の平均を求める
ID Finished Event	全てのワーカーからMES.OBJECT_3Dを送信した
Succeed Object ID Event	エージェントシー内での対象同定が成功した
Fail Object ID Event	エージェントシー内での対象同定が失敗した
No 3Dp Event	全てのワーカーに関して対象同定失敗した
Bye Agency Event	エージェントシーから脱退する
Nominate Event	マスタをノミネートする
Accept Nomination Event	マスタからのノミネートに反応する
Refuse Nomination Event	マスタからのノミネートを拒否する
Spawn Agency Event	Agency Spawningを行う
Succeed Agency ID Event	エージェント間での対象同定が成功した
Fail Agency ID Event	エージェント間での対象同定が失敗した
Restruct Agency Event	Agency Restructuringを行う
Trans AVA Event	AVAの移居要求に応じて移居領域を決定する
No Trans AVA Event	AVAの移居要求を拒否する
Accept Candidate Event	移居候補のAVAをエージェントシーに参加させる
Refuse Candidate Event	移居候補のAVAをエージェントシーに参加させない
Unify Agency Event	Agency Unificationを行う
Amalgam Agency Event	Agency Amalgamationを行う
Segregate Agency Event	Agency Segregationを行う
Same With ID Failure Event	Segregateされるエージェントのマスタになる
Differ From ID Failure Event	現在所属するエージェントのマスタであり続ける

表 2: イベントの種類

3 AVA 内におけるモジュール間動的インタラクション

3.1 AVA の3つの機能とAVAの状態

知覚機能は、対象の色、形、位置といった実世界の環境情報を観測するものである。本システムでは、カメラにより撮影される画像から、知覚結果として次のような情報を取り出して使用する。

- 画像中の検出領域の数
- 各検出領域の重心
- 画像上の検出領域
- 画像上での検出画素数
- カメラの投影中心から各検出領域の重心までの視線ベクトル

行動機能は、自らの物理的な身体を動かす機能である。身体を動かして環境に対して働きかけることにより、実世界内の対象を誘導し、検出しやすくすることといったことが可能である。しかし、本稿で設計するシステムにおける行動機能は、いまだ環境に対して大きな影響を与えるほどにはいたっていない。現在では、AVAの状態を調べて、適切なカメラパラメータを設定することと、カメラの視線方向を変更する機能に

限定されている。カメラパラメータは次のような内容を持っている。

- カメラのパンの値
- カメラのチルトの値
- カメラのズームの値

通信モジュールはコンピューターネットワークを介して情報の交換を行うことにより協調を行う。

このうち、対象追跡におけるAVAの自律性は主に知覚モジュールと行動モジュールのインタラクションによって実現されている。知覚モジュールと行動モジュールを動的に統合するものとしてダイナミックメモリを用いる。また、通信モジュールに関しては、メッセージバッファを用いたメッセージ交換を行い、得られた対象の位置情報をダイナミックメモリに書き込むことにより、行動モジュールにより正確な対象の位置を伝える。

AVAには大きく分けてマスタ、ワーカー、フリーランサーという状態が存在する。さらに、各状態は、モジュールやシステムの動作状態によって様々な副状態を取りながら動作する。AVAは検出対象の有無、受け取ったメッセージ、そしてAVAの動作結果とともに生成されるイベントによってこれらの副状態を遷移してゆく。状態遷移の詳細に関しては6で説明する。

3.2 ダイナミックメモリによるモジュールの統合

知覚、行動各モジュール間で情報交換を行うためには、扱う情報に共通の基盤を設けなければならない。具体的には、相手の情報がどのような内容で、いつ生成された情報なのか、ということを決定しなければならない。各モジュールから出力される情報を有効に用いるためには、その情報が生成された時刻が重要となる。各モジュールは固有の時間軸を持っており、情報生成の時刻はその時間軸にしたがって決定される。この時間軸がモジュール間で異なっていれば、モジュール間での情報交換はうまくいかない。よって、ここでは、各モジュールの時間軸は、そのモジュールの動作している計算機の時刻を参照するものとする。また、複数の計算機間では時刻は一致しているものと仮定する。モジュール間で情報交換を行う際に、モジュール間で直接情報交換を行うということをする、必然的に相手の情報を待つ、つまり、データを介したモジュール間の同期を取るということをしなければならない。並行・自律的に動作している各モジュールにとって、相手のデータを待つという動作は自分の動作を制限する制約となる。この制約をなくすために、本稿で述べるシステムではダイナミックメモリアーキテクチャを用いている。ダイナミックメモリとは、時間的に拡張された共有メモリである。ダイナミックメモリに記録されたデータを各モジュールが参照することにより、モジュール間の直接参照を避けることができる。ダイナミックメモリに記録される情報には以下のような特徴がある。

- 各変数は時間的に離散的なデータ系列を値として持つ。
- データは時系列にそって格納される。
- 複数の処理主体からいつでも、いつのデータでも自由に読み書きできる。

ダイナミックメモリに記録されるデータは、ダイナミックメモリにより時間的に拡張され、時間的に連続なデータとして展開される。具体的には、時刻 t_n にダイナミックメモリに記録された情報を $INFO(t_n)$ とすると、ダイナミックメモリ内で時間に関する補間関数 $f(t)$ が区間 $[t_n, t_{n+1})$ で定義され、離散情報 $INFO(t_n)$ は時間的に連続な $INFO(t)$ へと拡張される (式 (1))。

$$INFO(t) = f(t) : [t_n, t_{n+1}) \quad (1)$$

さらに、式 (1) を最新の情報に対して区間 $[t_n, \infty)$ で用い、 $f(t)$ を予測関数として用いることにより、未来の時刻の情報を予測することが可能である。これらの補間関数、予測関数には記録する情報の種類によってモデル化された関数を用いるべきであるが、本稿では、情報の補間には線形補間を用いる。各モジュールはダイナミックメモリに他のモジュールとは非同期にアクセスでき、さらに変数の任意の時刻の値を読み書きできる。それによってモジュール間のデータを介した同期が不必要になる。つまり、ひとつのモジュールは、他のモジュールの動作、状態などと関係なく必要なデータを得て動作することができる。

ダイナミックメモリとメッセージバッファを用いることにより、知覚、行動、通信の各機能は動的に統合される。

4 Agency 内における AVA 間動的インタラクション

4.1 対象の同定

対象の同定を行うためには最低でも2つの AVA が対象を捕えている必要がある。対象を検出した AVA には Detect Object Event が発生する。Detect Object Event が発生すると、通信モジュールは AVA の状態を調べ、自分がワーカであれば検出した対象情報をマスタに報告する。ワーカからのメッセージを受け取ったマスタは、その情報をもとに対象の同定を行うが、この時に、ワーカから受け取ったメッセージと現在の時刻との差が大きければ、たとえ同じ対象を検出していても対象の同定は成功しない。そのため、ダイナミックメモリを用いて受け取ったメッセージに含まれる検出情報(カメラから画像上の対象の重心へ向かうベクトル)と自分の持つ検出情報の時刻差を吸収するというを行う。ダイナミックメモリを用いて2つの情報の時刻差を吸収することを、仮想同期 [4] と呼ぶ。以下で、仮想同期を用いた対象位置の算出と対象同定について説明する。ワーカからのメッ

セージを待っているマスタは、メッセージ待ちのタイムアウトまでの間に以下の操作を繰り返す。

- ワーカからのメッセージから、その AVA から対象までの視線方向ベクトルを取りだし、ダイナミックメモリに記録する。
- ワーカからの対象情報と、自分の持つ対象情報について、現在の値を検索する (図 3)。
- 時刻が一致した2つの視線情報(空間中の線分情報)をもとに、両方の線分に対してもっとも近い3次元点を算出する。

全てのワーカからのメッセージを処理してしまうか、メッセージ待ちがタイムアウトした後に、複数算出された3次元点の平均の位置を算出し、これをエージェントが追跡している対象の位置であるとする。そして、対象の位置から各視線ベクトルまでの最短距離が一定値以内の AVA は対象同定が成功したものとし、一定値以上距離が離れている AVA に関しては対象同定が失敗したものとす。

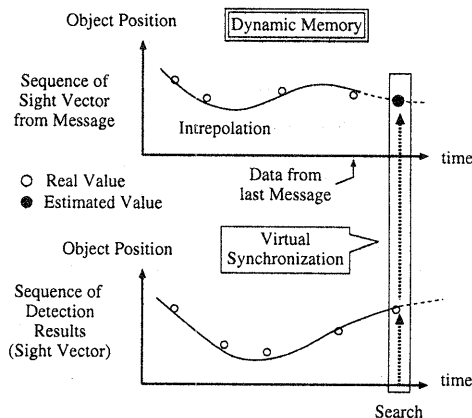


図 3: 仮想同期

4.2 インタラクションのプロトコル

Agency Formation エージェントが存在しない場合に対象を検出した AVA はエージェントを形成する。エージェントが存在する場合には、対象を検出した AVA はマスタに対して対象同定のための問い合わせを行う。対象が同定されれば、ワーカとしてエージェントに参加し、同定が失敗すれば、自らマスタとして新たなエージェントを形成する。AVA は対象を見失ったり、エージェントのマスタからエージェント変更などの命令を受け取ると、AVA はエージェントから脱退する。

Role Assignment 対象は独自のダイナミクスに従い動作するため、システムはそのダイナミクスに適応的に処理することが要求される。そのためには、エージェントのマスタは常に対象を安定して捕らえていることが望ましい。ここで、エージェント内のマスタを動的に入れ替える

ということを行う。エージェントのマスタは、対象同定の際にワーカのカメラ情報、検出情報などを得ることができ、それらをもとにワーカのエージェントに対するユーティリティを計算し、その値に従い次のマスタを選択する。次のマスタはメッセージによってエージェントのメンバ全員に通知される。ユーティリティは検出画像中の対象サイズや、検出情報の新しさなどにより決定される。このようにマスタを入れ替えることにより、対象の動きや場所に関わらず、エージェントを安定に存在させることができる。マスタを入れ替えるに当たり、次のマスタ候補のAVAからマスタになれるかどうかという返答を受け取らなければならないため、ここでは同期通信が行われる。

Agency Spawning もともと一つとして検出されていた対象が、何らかの理由により分離した場合には、対象を2つとして検出したAVAはエージェントのマスタに2つの対象情報を通知する。マスタは、その対象のうち、対象同定の失敗した方について、エージェントが存在するかどうかを確かめた後、エージェントが存在しなければAVAを新エージェントのマスタとして独立させる。こうして、一つのエージェントから新たなエージェントが生成され、二つに分離するということが起る。

5 Agency 間の動的インタラクション

対象が複数存在する場合にはエージェントも複数生成される。対象が単数であると仮定できる場合は、対象の動きの多様性に対応できるようにシステムを設計すればよいが、対象が複数の場合には、対象の数の多様性についても考慮する必要がある。具体的には、対象が複数であることにより、次のような問題が生じる。

1. 対象同定の失敗などにより、エージェントの数と対象の数がずれてしまう。
2. 複数の対象が近接した場合に、それらを分離して検出することができない。
3. 多数のAVAに追跡されている対象と、少数のAVAにしか追跡されていない対象ができる。

対象が複数であることによって生じる問題に対してはエージェント間のコミュニケーションによって解決をはかる。エージェント間のコミュニケーションは各エージェントのマスタ間の通信によって行われる。

5.1 インタラクションプロトコル

Agency Unification 対象同定のタイミングのずれや、対象の位置計算の失敗により、一つの対象に関して複数のエージェントができることが

ある。この場合には、エージェントを統合して一つのエージェントにまとめる必要がある。エージェントは定期的に自分の追跡している対象の位置情報などをブロードキャストしている。それをもとに、エージェント間で対象の同定を行い、同定された場合には同じ対象を追跡していたものと判断し、複数のエージェントを統合する。

Agency Amalgamation 複数の対象がすれ違う場合には、検出アルゴリズムによって対象が分離検出できない場合がある。2つの対象についてエージェント間での対象同定が成功した場合には、両対象の位置変化の履歴を調べ、別の対象であることを確認した後、一時的にエージェントが融合される。また、Agency Amalgamation が起ると、対象同定が不安定な状態としてエージェントは Amalgam 状態であるとする。それ以外の場合はエージェントは Normal 状態であると大別する。

Agency Segregation Agency Amalgamation によって、エージェントが融合されると、エージェントは通常の動作よりもエージェントを元のように分離することを優先する。一つとして検出されていた対象が二つに分離すると、エージェントを元のように分離して追跡を継続する。

Agency Restructuring 複数のエージェントのメンバ数に大きな差がある場合、AVAの要求が行われる。具体的には、エージェント間で対象の同定を行う際にエージェントのメンバ情報を交換し、AVAの数に格差がある場合に、AVAの要求が出される。要求を受けたエージェントのマスタは、メンバのユーティリティを元に移籍候補AVAを選び、その候補情報を要求元に伝える。要求元のエージェントは候補AVAの情報をみて、受け入れるかキャンセルするかを決定する。AVAを移籍させるに当たり、候補情報と受け入れるかどうかの返答をやり取りしなければならないのでここでは同期通信が行われる。Agency Restructuringを行うことにより、特定のエージェントにAVAが片寄ることを防ぐことができる。

6 AVAの状態遷移の設計

本章では、これまで述べてきたAVAの動作を記述した状態遷移図を用いて、

1. 4, 5章でそれぞれ説明したエージェント内、エージェント間で起こるAVA間の動的インタラクションに相当する部分の動作。
2. 本システムの特徴的動作を示す具体例。

について述べる。AVAの役割がフリーランサー・ワーカの場合、マスタ状態における状態遷移図を、それぞれ図4¹, 5に示す。状態遷移を引き起こす要素²は、

¹ 図中下部に各状態・遷移を表す記号の意味を記載してある。

² 図中で、状態遷移を示す矢印の上に記されている。

した場合、エージェントが Amalgam 状態の時
に Master(IDFail) 状態に遷移した場合。

6.2 システムの動作の具体例

本節では、1) 通常は即応性を重視して非同期に動作している AVA が、同期を必要とする、2) 本システムで複数対象追跡を行う際に必要になるエージェント間の交渉を必要とする、の両方の特性をあわせ持った Agency Restructuring の実行を具体例に挙げて、複雑なインタラクションを伴うシステムの動作を確認する。

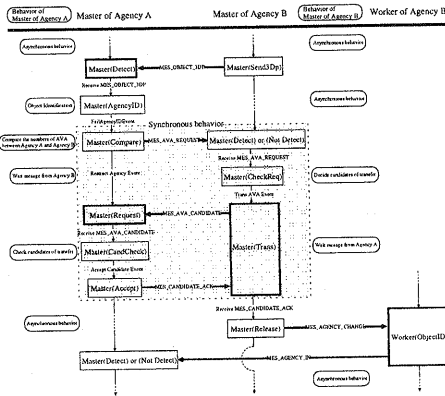


図 6: Agency Restructuring 時の動作

図 6 は、Agency Restructuring に関する AVA の動作を表した図である。各状態における AVA 間のインタラクションと、同期・非同期動作が切り替わるタイミングについて以下で述べる。

1. **Agency A** エージェント A のマスタ (以降, AVA_{AM}) がエージェント B のマスタ (以降, AVA_{BM}) からブロードキャストされた MES.OBJECT_3DP を受信すると、追跡対象の同定を行う。同定が失敗し、且つ AVA の移籍要求が決定されると、MES.AVA_REQUEST を AVA_{BM} に対して送信した後、AVA_{BM} からの返信待ち状態に移る。即ち、以降同期動作に移る。
2. **Agency B** AVA_{BM} が MES.AVA_REQUEST を受信し、且つその要求に応えるならば、移籍候補を決定し MES.AVA_CANDIDATE を返信した後、AVA_{AM} からの返信待ち状態に移る (同期動作)。
3. **Agency A** AVA_{AM} が MES.AVA_CANDIDATE を受信し、且つその移籍候補を受け入れるならば、MES.CANDIDATE_ACK を返信し、以降通常の非同期的動作に復帰する。
4. **Agency B** AVA_{BM} が MES.CANDIDATE_ACK を受信すると、移籍が決定した AVA に対し MES.AGENCY_CHANGE を送信し、以降通常の非同期的動作に復帰する。

7 おわりに

本論文では、動的に変化する実世界を把握することを目的とするシステムの構築をテーマにした。実世界を把握する上で、実世界内に存在する可動物体の位置を調べ、追跡するということは重要である。本研究では、実世界把握に有効であろうと考えられる視覚的情報を用いて、実世界内の複数対象の追跡を行うシステム的设计を行った。動的に変化する実世界を捕らえる上で、集中管理型のシステムでは設計時の想定外の事態に対応できない、実世界のスケールに合わせてシステムの規模を変更できないということがある。よって、実世界を対象とする上で必要となる実時間性、適応性などを実現するために、自律的に動作する能動視覚エージェントを複数台環境内に配置し、それらの間の協調処理によって実現するという分散協調型のシステムを設計するという立場をとった。

本研究では、自律的に動作する処理主体として AVA を用いた。AVA は環境情報を取得する知覚、物理的身体を動かす行動、そして他の AVA と通信する通信というモジュールを持っており、これらのモジュールも自律的に動作することができる。また、AVA の集合としてのエージェントを考え、各モジュール間の動的インタラクション、AVA 間の動的インタラクション、そしてエージェント間の動的インタラクションという 3 層の処理構造を設定した。モジュールを基本とした層では、ダイナミックメモリを用いて知覚、行動、通信各モジュールを統合し、AVA を基本とした層では、マスタ、ワーカを主体とした、メッセージの交換を基礎とした協調プロトコルを用いて対象の追跡を実現し、エージェントを基本とした層では、エージェント間の協調プロトコルによってシステムのリソースバランスの調整、検出アルゴリズムの不完全さを補うということを行った。

本研究は、日本学術振興会未来開拓学術研究推進事業 (JSPS-RFTF 96P00501) の補助を受けて行った。

参考文献

- [1] T.Matsuyama: Cooperative Distributed Vision - Dynamic Integration of Visual Perception, Action, and Communication -, *Proc. of Image Understanding Workshop*, pp. 365-384, Monterey CA, (1998).
- [2] 村瀬 健太郎: 実時間対象検出・追跡のための知覚と行動の動的統合. 修士論文, 京都大学工学研究科電子通信工学専攻, 1999.
- [3] 松山隆司, 和田俊和, 丸山昌之: 能動視覚エージェントによる移動対象の協調的追跡, 画像の認識・理解シンポジウム (MIRU'98) 論文集 I, pp. 365-370 (1998).
- [4] 吉岡章夫: 複数エージェントの仮想的同期に基づく協同注視システム. 修士論文, 京都大学工学研究科電子通信工学専攻, 1999.